



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/865,001	05/24/2001	Marc Noel Blais	ROC9-2000-0162-US1	3815

46296 7590 03/30/2005

MARTIN & ASSOCIATES, LLC
IBM INTELLECTUAL PROPERTY LAW DEPARTMENT
DEPARTMENT 917, BUILDING 006-1
3605 HIGHWAY 52 NORTH
ROCHESTER, MN 55901-7829

EXAMINER

ALI, SYED J

ART UNIT	PAPER NUMBER
----------	--------------

2195

DATE MAILED: 03/30/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/865,001	Applicant(s) BLAIS ET AL.	
	Examiner Syed J Ali	Art Unit 2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 28 January 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,3-6,8-12 and 15-18 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1,3-6,8-12 and 15-18 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This office action is in response to the amendment filed January 28, 2005. Claims 1, 3-6, 8-12, and 15-18 are presented for examination.

2. The text of those sections of Title 35, U.S. code not included in this office action can be found in a prior office action.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. **Claim 3 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.**

5. Claim 3 recites the limitation "The apparatus of claim 2" in line 1. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 103

6. **Claims 1, 3-6, 8-12, and 15-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Whaley ("Compositional Pointer and Escape Analysis for Java Programs") (hereinafter Whaley) in view of Choi et al. ("Escape Analysis for Java") (hereinafter Choi).**

7. As per claim 1, Whaley teaches the invention as claimed, including an apparatus, comprising:

at least one processor (Abstract);

a memory coupled to the at least one processor (Abstract);

a first compilation unit residing in the memory, the first compilation unit comprising a plurality of object oriented classes that are part of an object oriented program, wherein the object oriented program is defined by the combination of the first compilation unit and at least one other compilation unit (§§ 3.1, 3.2); and

a compiler residing in the memory and executed by the at least one processor (§1.2) in a partial compilation environment (§8.1), the compiler allocating at least one object in the first compilation unit to an invocation stack frame for a method in the first compilation unit that allocates the at least one object (§7.2).

8. Choi teaches the invention as claimed, wherein the compiler comprises:

an escape analysis mechanism that marks each instruction in the first compilation unit that allocates a new object as one of global escape, no escape, and arg escape based on information available from classes visible in the first compilation unit (§2.1, Proposition 2.3); and

an object allocation mechanism that allocates at least one object that is created by an instruction marked as no escape by the escape analysis mechanism to an invocation stack frame for a method that allocates the object (§2.1, Proposition 2.3).

Art Unit: 2195

9. It would have been obvious to one of ordinary skill in the art to combine Whaley and Choi since the markings provided by Choi provides a simple data model for the extensive analysis procedures performed by Whaley (Choi, Abstract; Whaley, §§1-1.2).

10. As per claim 3, Choi teaches the invention as claimed, including the apparatus of claim 2 wherein the escape analysis mechanism marks each instruction in the first compilation unit that allocates a new object as one of global escape, no escape, and arg escape based on information available from classes visible in the first compilation unit (§2.1, Proposition 2.3) and from classes that are outside the first compilation unit that are visible in a specified classpath (§§2.1, Proposition 2.3; 4).

11. As per claim 4, Whaley teaches the invention as claimed, including an apparatus, comprising:

at least one processor (Abstract);

a memory coupled to the at least one processor (Abstract);

a first compilation unit residing in the memory, the first compilation unit comprising a plurality of object oriented classes that are part of an object oriented program, wherein the object oriented program is defined by the combination of the first compilation unit and at least one other compilation unit (§§ 3.1, 3.2); and

Art Unit: 2195

a compiler residing in the memory and executed by the at least one processor (§1.2) in a partial compilation environment (§8.1), the compiler allocating at least one object in the first compilation unit to an invocation stack frame for a method in the first compilation unit that allocates the at least one object (§7.2), wherein the compiler comprises:

a code generator that creates two versions of code for a selected object method, a first version using stack allocation of objects (§7.2) and a second version using heap allocation of objects (§§1.1, 7.2); and

a run time code selector that selects one of the first and second versions to execute at run time based on a determination of whether classes seen at run time match expected classes within predetermined limits (§§1.2, 7.2, 8.1).

12. As per claim 5, Whaley teaches the invention as claimed, including an apparatus comprising:

at least one processor (Abstract);

a memory coupled to the at least one processor (Abstract);

a first compilation unit residing in the memory, the first compilation unit comprising a plurality of object oriented classes that are part of an object oriented program, wherein the object oriented program is defined by the combination of the first compilation unit and at least one other compilation unit (§§ 3.1, 3.2); and

a compiler residing in the memory and executed by the at least one processor (§1.2) in a partial compilation environment (§8.1), the compiler comprising:

an escape analysis mechanism that operates on the first compilation unit prior to a second compilation unit (§8.1) and that performs an escape analysis in the first compilation unit that allocates a new object based on information available from classes visible in the first compilation unit but not visible in the uncompiled second compilation unit (§§ 1.2, 8.1);

a code generator that creates two versions of code for a selected object method, a first version using stack allocation of objects (§7.2) and a second version using heap allocation of objects (§§1.1, 7.2); and

a run time code selector that selects one of the first and second versions to execute at run time based on a determination of whether classes seen at run time match expected classes within predetermined limits (§§1.2, 7.2, 8.1).

13. Choi teaches the invention as claimed, including the compiler further comprising:

an escape analysis mechanism that marks each instruction in the first compilation unit that allocates a new object as one of global escape, no escape, and arg escape based on information available from classes visible in the first compilation unit and from classes that are outside the first compilation unit that are visible in a specified classpath (§2.1, Proposition 2.3); and

an object allocation mechanism that allocates at least one object that is created by an instruction marked as no escape by the escape analysis mechanism to an invocation stack frame for a method that allocates the object (§2.1, Proposition 2.3).

Art Unit: 2195

14. As per claim 6, Whaley teaches the invention as claimed, including a method for allocating objects to memory in an object oriented program that comprises a first compilation unit and at least one other compilation unit (§§3.1, 3.2), the method comprising the steps of:

- (A) compiling the first compilation unit (§1.2); and
- (B) during the compiling of the first compilation unit and before the compilation of the second compilation unit (§8.1), performing an escape analysis based on information available from classes in the first compilation unit (§2.1, Proposition 2.3); and

15. Choi teaches the invention as claimed, including:

- (B) during the compiling of the first compilation unit, marking each instruction that allocates a new object as one of global escape, no escape, and arg escape based on information available from classes in the first compilation unit and from classes that are outside the first compilation unit that are visible in a specified classpath (§§ 1.2, 8.1); and
allocating at least one object that is created by an instruction marked as no escape by the escape analysis mechanism to an invocation stack frame for a method that allocates the at least one object (§2.1, Proposition 2.3).

16. As per claim 8, Whaley teaches the invention as claimed, including the method of claim 6 wherein step (B) comprises the steps of:

creating two versions of code for a selected object method, a first version using stack allocation of objects (§7.2) and a second version using heap allocation of objects (§§1.1, 7.2); and

Art Unit: 2195

selecting at run time one of the first and second versions to execute at run time based on a determination of whether classes seen at run time match expected classes within predetermined limits (§§1.2, 7.2, 8.1).

17. As per claim 9, Whaley teaches the invention as claimed, including in an object oriented computer program that comprises a first compilation unit and at least one other compilation unit, a method for allocating objects in the first compilation unit to memory, the method comprising the steps of:

creating two versions of code for a selected object method, a first version using stack allocation of objects (§7.2) and a second version using heap allocation of objects (§§1.1, 7.2); and

selecting at run time one of the first and second versions to execute at run time based on a determination of whether classes seen at run time match expected classes within predetermined limits (§§1.2, 7.2, 8.1).

18. Choi teaches the invention as claimed, including the method further comprising:

marking each instruction that allocates a new object as one of global escape, no escape, and arg escape based on information available from classes in the first compilation unit and from classes that are outside the first compilation unit that are visible in a specified classpath (§2.1, Proposition 2.3).

19. As per claim 10, Whaley teaches the invention as claimed, including a program product comprising:

Art Unit: 2195

a compiler that compiles in a partial compilation environment (§8.1) a first compilation unit comprising a plurality of object oriented classes that are part of an object oriented program (§1.2), wherein the object oriented program is defined by the combination of the first compilation unit and at least one other compilation unit (§§ 3.1, 3.2), the compiler allocating at least one object in the first compilation unit to an invocation stack frame for a method in the first compilation unit that allocates the at least one object (§7.2); and

signal bearing media bearing the compiler (Abstract).

20. Choi teaches the invention as claimed, wherein the compiler comprises:

an escape analysis mechanism that marks each instruction in the first compilation unit that allocates a new object as one of global escape, no escape, and arg escape based on information available from classes visible in the first compilation unit (§2.1, Proposition 2.3); and

an object allocation mechanism that allocates at least one object that is created by an instruction marked as no escape by the escape analysis mechanism to an invocation stack frame for a method that allocates the object (§2.1, Proposition 2.3); and

wherein the escape analysis mechanism marks each instruction in the first compilation unit that allocates a new object as one of global escape, no escape, and arg escape based on information available from classes visible in the first compilation unit (§2.1, Proposition 2.3) and from classes that are outside the first compilation unit that are visible in a specified classpath (§§2.1, Proposition 2.3; 4).

Art Unit: 2195

21. As per claim 11, Whaley teaches the invention as claimed, including the program product of claim 10 wherein the signal bearing media comprises recordable media (Abstract).

22. As per claim 12, Whaley teaches the invention as claimed, including the program product of claim 10 wherein the signal bearing media comprises transmission media (Abstract).

23. As per claim 15, Whaley teaches the invention as claimed, including the program product of claim 10 wherein the compiler comprises:

a code generator that creates two versions of code for a selected object method, a first version using stack allocation of objects (§7.2) and a second version using heap allocation of objects (§§1.1, 7.2); and

a run time code selector that selects one of the first and second versions to execute at run time based on a determination of whether classes seen at run time match expected classes within predetermined limits (§§1.2, 7.2, 8.1).

24. As per claim 16, Whaley teaches the invention as claimed, including a program product comprising:

(A) a compiler that compiles a first compilation unit comprising a plurality of object oriented classes that are part of an object oriented program (§1.2), wherein the object oriented program is defined by the combination of the first compilation unit and at least one other compilation unit (§§ 3.1, 3.2), the compiler comprising:

Art Unit: 2195

- (A3) a code generator that creates two versions of code for a selected object method, a first version using stack allocation of objects (§7.2) and a second version using heap allocation of objects (§§1.1, 7.2); and
 - (A4) a run time code selector that selects one of the first and second versions to execute at run time based on a determination of whether classes seen at run time match expected classes within predetermined limits (§§1.2, 7.2, 8.1); and
 - (B) signal bearing media bearing the compiler (Abstract).
25. Choi teaches the invention as claimed, including the compiler further comprising:
- (A1) an escape analysis mechanism that marks each instruction that allocates a new object as one of global escape, no escape, and arg escape based on information available from classes in the first compilation unit and from classes that are outside the first compilation unit that are visible in a specified classpath (§2.1, Proposition 2.3); and
 - (A2) an object allocation mechanism that allocates at least one object that is created by an instruction marked as no escape by the escape analysis mechanism to an invocation stack frame for a method that allocates the object (§2.1, Proposition 2.3).
26. As per claim 17, Whaley teaches the invention as claimed, including the program product of claim 16 wherein said signal bearing media comprises recordable media (Abstract).

27. As per claim 18, Whaley teaches the invention as claimed, including the program product of claim 16 wherein said signal bearing media comprises transmission media (Abstract).

Response to Arguments

28. **Applicant's arguments filed January 28, 2005 have been fully considered but they are not persuasive.**

29. Applicant argues, *"The cited sections of Choi (and apparently the whole document) are concerned with escape analysis in a full compilation environment. Nothing in Choi teaches or suggests to apply escape analysis in a partial compilation environment."*

30. I should be noted that Whaley is cited as teaching partial compilation. Whaley discusses performing escape analysis in a dynamic compiler independently of application compilation (§1.2). The escape analysis is performed in a distinct compilation phase, such that certain libraries or components may be separately analyzed to save time during compilation of a larger program (§8.1). Whaley discusses escape analysis in a partial compilation environment extensively, highlighting the advantages of performing the analysis and compilation in distinct phases.

Choi is cited as presenting a way to mark the escapability of objects using a simple data model (§2.1, Proposition 2.3). Choi also indicates that the escape analysis can be used in a number of distinct compilation environments, and is not limited to a full compilation environment (§1). The escape analysis can be used in static or **dynamic** compilers, Java

Art Unit: 2195

application extractors, or bytecode optimizers (§1) (emphasis added). Thus, Choi is suitable for a partial compilation environment, such as that presented by Whaley.

31. Applicant argues that Whaley “*describe[s] escape analysis and the compiler allocating to the stack or heap, but there is no discussion concerning the code generator (compiler) making two versions of code.*”

32. Examiner respectfully disagrees. Whaley discusses generating two versions of code, one which is suitable for heap allocation if an object does not escape a thread (§7.1) and a second that is allocated to the heap when an object does not escape a method (§7.2). The objects are partially compiled, i.e. analyzed as a separate phase of the compilation process, and the appropriate allocation is made thereafter (§8.1). When performing compilation on an application, the analysis is used to apply the allocation (§8.1).

33. Applicant argues that Choi does not teach “*marking a new instruction...based on information...that are visible in a specified classpath.*” Applicant argues that “*a specified classpath can be on specified by the user.*”

34. The feature that a specified classpath can only be specified by a user is conspicuously absent from the claims. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Classpaths are presented as indicating a set of visible classes that are outside the compilation unit. Choi teaches a method of tracing the reachability of objects to determine a caller-callee relationship (§4). This functions as a means

Art Unit: 2195

of generating a call graph, such that the escape analysis exhaustively determines the escapability of objects by determining the full interprocedural reach of an object.

Conclusion

35. Applicant's amendment necessitated the new grounds of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (571) 272-3769. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Art Unit: 2195

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Syed Ali
March 23, 2005



MENG AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100